Remarks

Responsive to the final Office Action mailed August 26, 2005, Applicants provide the following remarks. Reexamination and allowance of the subject application is respectfully requested.

Previously pending claims 1-34 have been cancelled in favor of new claims 35-50. Support for the new claims may be found, for example, in FIGS. 1 through 5, the description on pages 2 through 6, the first full paragraph beginning on page 7, the first full paragraph on page 13, and pages 14 through 18, etc. No new matter is believed entered by the new claims.

Applicants respectfully submit that the rejections of claims 1-34 are rendered moot by the cancellation thereof.

With regards to the new claims, Applicants respectfully submit that the references cited in the final Office Action fail to teach or suggest every aspect of the invention defined therein. Among other limitations, the new claims require, for example, providing a clock capable of indicating a current time and, as a separate element, providing a timer. As such, the timer is a discrete element from the clock. Additionally, the claims require, in part, determining a time period based on a difference between a start time, which is based on the time when priority of expiration times is determined, and utilizing a timer to time the time period. The foregoing, and other, aspects of the new claims are believed to distinguish over the cited references.

By contrast to the limitations of the new claims, Cave (US 6,232,808) is understood to teach a method of timing multiple events without the use of a clock and timer as separate elements. For example, Cave discloses:

> "According to the present invention, a single compare register is also associated with the clock register, the compare register preferably being of equivalent length to the clock register. The compare register may be a separate register or the bottom register in a "stack" of registers configured in either hardware or software. Depending on the embodiment

> deployed, hardware or memory stores chronologically ordered timing values and supplies them in sequence to the compare register. A comparator monitors the clock register's current value and compares it with the timing value currently loaded in the compare register. As the clock register's value reaches the current timing value in the compare register, an alert signal (e.g., a processor interrupt or some other reason for setting a timing value) is generated and sent out with a corresponding event ID ("EID") associated with the timing value in the compare register. The current timing value in the compare register is then discarded, and the next timing value is loaded into the compare register." (Col. 3, l. 5-24, emphasis added)

Accordingly, rather than setting a timer corresponding to any time period, Cave teaches to simply continuously compare a current time, as provided by a continuously running clock, to a timing value. See, e.g., col. 7, l. 25-30. As such, Cave does not teach providing a clock and separately providing a timer. In fact, the teachings of Cave are inconsistent with such an approach, as the method of Cave eliminates the need for a timer, by simply comparing a timing value with a current time on a continuous, ongoing basis.

Similar to Cave, Devanagundy et al. (US 6,002,737) teach an approach to timing multiple time-outs that only utilizes a free-running counter. Devanagundy et al. is understood to teach a timing approach that includes comparing a time elapse, i.e., the time between a current time and a start time, against a desired time interval, i.e., the timeout values indicating the length, in counter clock cycles, of the time-out period. For example, Devanagundy et al. disclose:

> "In accordance with the invention, a circuit such as a host adapter includes a timer capable of detecting time-outs for multiple pending commands that have different time-out periods. The timer includes a single free-running counter, a first storage for start counts, a second storage for time-out values, a subtractor, and a comparator. The start counts are counts from the free-running counter that the host adapter saves to the first storage when issuing an associated command. The timeout values indicate the lengths in counter clock cycles of different types of time-out periods. To check whether a command timed out, the host adapter selects from the first storage a start count associated with the command, and the subtractor determines a difference between a current

> count in the free-running counter and the selected start count. The host also selects from the second storage a time-out value according to the type of time, and the comparator compares to the selected time-out value to the difference from the subtractor. The comparator asserts a signal indicating the command timed out if the difference from the subtractor is greater than the selected time-out value. Thus, a timer with a single comparator can monitor many types or time-out for many instructions." (Col. 1, l. 62 through col. 2, l. 16, emphasis added)

Accordingly, Devanagundy et al. also teach an approach that does not include a clock and timer provided as separate elements. In fact, as with Cave, the disclosed approach would not even be susceptible to the use of a clock and timer, as the timer would not serve a purpose in the context of the disclosed method.

Finally, looking to Short et al. (US 5,708,814), Applicants first note that Short et al. do not teach an approach to timing multiple events. Rather, Short et al. teach a method for allowing multiple interrupts to accumulate and to be serviced by a CPU as a batch, rather than individually. For example, Short et al. disclose:

> "In accordance with the present invention, a peripheral device interrupt controller controls the rate at which interrupts to the CPU by a peripheral device are generated such that a number of peripheral device operations which otherwise would individually initiate an interrupt are effectively serviced as a "batch" by the CPU with a single interrupt. The controller includes a timer for timing a delay from an earliest pending peripheral device operation, and a counter for counting the number of pending peripheral device operations. When the timer or the counter reach preset limits, the controller generates an interrupt to the CPU. The preset limits may be set by software. On receiving the interrupt, the CPU preferably services all pending peripheral device operations. As a result, multiple peripheral device operations which occur in a burst are processed in a single interrupt by the CPU. This dramatically reduces the interrupt processing overhead of the CPU (i.e., the time spent saving and restoring its current execution state) in a busy system, and thereby increases both the throughput and responsiveness of the system. Further, the rate at which the peripheral devices generate interrupts can be placed under software control." (Col. 1, l. 51 through col. 2, l. 4, emphasis added)

As indicated above, Short et al. teach the use of a delay timer which may allow additional interrupt events to accumulate and which may allow batch servicing of interrupts, rather than individual servicing of interrupt events. The delay timer does not time an event, but merely runs for a preset delay time which is not understood to vary with, or be in anyway dependent upon, the event. In this regard, Short et al. further disclose:

> "The delay timer 62 times an interval determined by the preset delay time value in the delay time register 57 from the earliest interrupt event still pending or the last interrupt event serviced (whichever is later). This has the effect of both imposing a delay from an earliest interrupt event during which additional interrupt events that occur in a burst with the earliest interrupt event can accumulate and be serviced as a batch, as well as a minimum delay between the CPU 24 servicing interrupt events of the controlled peripheral device and the interrupt controller 50 asserting a next IRQ for the controlled peripheral device." (Col. 4, l 65 through col. 5, l. 9, emphasis added)

In sum, Applicants respectfully submit that there is no teaching, suggestion, or motivation that would lead one having ordinary skill in the art to consider the further teachings of Short et al. in combination with Cave and/or Devanagundy et al. to achieve an approach to timing multiple events. Short et al. is simply not relevant to timing multiple events.

Aside from the lack of any teaching, suggestion, or motivation to combine the teachings of Short et al. with the other cited references, as noted above, both Cave and Devanagundy et al. relate to timing approaches that seek to eliminate the use of a separate timer and clock. Accordingly, each of these references teaches away from the further inclusion of a timer.

Finally, Applicants respectfully submit that the further teachings of Short et al., even if combined with Cave and/or Devanagundy et al., fail to achieve the invention of the newly presented claims. As indicated above, Short et al. teach the use of a timer only for counting a predetermined time interval. As evident from the above quoted passages, Short et al. merely teach providing a preset time delay from the receipt of an interrupt

even, or a last interrupt event serviced, to allow additional interrupt events to accumulate. The accumulation of additional interrupt events may allow the multiple interrupt events to be serviced as a batch, rather than individually. As such, Short et al. do not teach timing multiple events. The time delay counted by the timer is not understood to vary with or by dependent upon the even. Rather the time delay controlled to be a preset time delay independent of an event.

In view of the foregoing, Applicants respectfully submit that the new claims herein patentably distinguish over the cited references. Accordingly, the application is believed to be in condition for allowance. An early allowance is respectfully solicited.

In the event of any fee deficiencies, or that additional fees are payable, please charge our Deposit Account No. 50-2121 as necessary.

Respectfully Submitted

RONALD D. OLSON ET AL.

By their Representatives,

Customer Number: 45459

Minneapolis, MN 55402

603-668-6560

Date: 7/26/06 By: _____

Edmund P. Pfleger
Reg. No. 41,252

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Attn-Mail Stop RCE, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this 26 day of JULY 2006.

Kyrstin Ryan
Name

Kyrstin Ryan
Signature